

## Informatica, 2019/2020 - Prova di teoria

Scrivete le risposte in modo chiaro, inequivocabile e a **penna**. Ogni risposta ambigua, non chiara, o incompleta sarà valutata nel modo più sfavorevole per lo studente. Tutte le domande a risposta multipla hanno esattamente una e una sola risposta corretta. Il compito verrà tenuto in considerazione solo se la risposta a tutte le domande marcate come **(Domanda di sbarramento)** sarà corretta. La pena per chi verrà sorpreso a copiare/suggerire è *l'annullamento immediato del compito ed il fallimento della prova*, più altre eventuali sanzioni a livello di corso e di ateneo. Dove non specificato, i logaritmi sono in base 2.

**Es 1. (Domanda di sbarramento)** Quale di queste affermazioni è falsa?

- A . il mergesort ha lo stesso tempo di esecuzione nel caso peggiore e nel caso migliore  
 B . il tempo di esecuzione di un algoritmo dipende solo dalla lunghezza dell'input  
 C . esistono istanze di input per cui il bubblesort risulta più veloce del mergesort  
 D . la stessa funzione può essere calcolata da algoritmi diversi

**Es 2. (Domanda di sbarramento)** Qual è la lista di indici generata dall'espressione python `list(range(3,8))`?

- A . []      B . [3,8]      C . [3,4,5,6,7]      D . [3,4,5,6,7,8]      E . [3,7]

**Es 3. (5 pt)** Cosa fa il seguente programma? Descrivete chiaramente cosa produce la funzione `mystery`, per qualunque valore intero. Fate anche un paio di esempi.

```
def mystery(n):
    if n < 1:
        return ""
    else:
        return "*" + mystery(n-1)
```

Rispondere qui

**Es 4. (5 pt)** Quale di queste relazioni asintotiche è vera?

- A .  $2^{(\log n)^2} \in O(n^2)$       B .  $2^{3\log(n)+4} \in O(n^3)$       C .  $n^{0.9} \log^5(n) \in \Omega(n)$       D .  $\sqrt{n} \in \Omega(4^{\frac{\log n}{2}})$

**Es 5. (3 pt)** Quali valori appariranno in *A* e in *B* dopo l'esecuzione delle seguenti istruzioni?

```
A = [1,0,1,0,1,0,1,0]
B = [1,1,1,1,1,1,1,1]
for i in range(0,len(A)):
    A[i] = A[i] * 2
    B[i] = B[i] + A[i//2]
```

A      0    1    2    3    4    5    6    7

B      0    1    2    3    4    5    6    7

**Es 6. (3 pt)** Qual è la lista di indici generata dall'espressione python `list(range(3,9,3))`?

- A . [3,4,5,6,7,8,9]      B . [3]      C . []      D . [3,6,9]      E . [3,6]      F . [3,4,5,6,7,8]

**Es 7. (6 pt)** Considerare l'invocazione `process(lista,0,len(lista))` della funzione `process` così definita:

```

def process(lista, start, end):
    if end-start < 2:
        return
    half = (end + start) // 2
    process(lista, start, half)
    process(lista, start, half)
    process(lista, half+1, end)
    process(lista, half+1, end)
    bubblesort(lista, start, end)

```

Dare l'equazione di ricorrenza che descrive il tempo di esecuzione  $T(n)$  di `process` nel caso peggiore, assumendo che  $n$  sia la lunghezza della lista:

Rispondere qui

Utilizzare inoltre il Master Theorem per determinare il tempo di esecuzione  $T(n)$ :

**A**□.  $T(n) \in \Theta(n^{2.5})$     **B**□.  $T(n) \in \Theta(n^{1.5})$     **C**□.  $T(n) \in \Theta(n^2 \log(n))$     **D**□.  $T(n) \in \Theta(n^3 \log(n))$

Disegnare infine l'albero della ricorsione per  $n = 4$ :

Rispondere qui

**Es 8. (4 pt)** Quali valori appariranno in  $B$  dopo l'esecuzione delle seguenti istruzioni?

```

A = [1,2,1,2,1,2,1,2]
B = [1,0,0,0,0,0,0,0]
for i in range(0,len(A)):
    B[i] = B[A[i]-1]*(i+1)

```

$B$     0    1    2    3    4    5    6    7

--	--	--	--	--	--	--	--	--

**Es 9. (4 pt)** Data una lista `seq=[-1,2,1,4,8,5,-12]` qual è il valore dell'espressione `-seq[1:-1][:-2][-1]`?

**A**□. -4    **B**□. Espressione non valida    **C**□. 12    **D**□. -8    **E**□. 2    **F**□. 4

**Es 10. (6 pt)** Disegnate l'albero di decisione corrispondente all'algoritmo bubblesort (ottimizzato con uso di flag al fine di evitare passate inutili) per 3 elementi.

Rispondere qui